
















Anhang zum Artikel „Unterrichtsreihe zu künstlicher Intelligenz und künstliche neuronale Netzen für die gymnasiale Oberstufe“

Katrin Gisela Maria Grabe

1) Arbeitsblatt „Berechnung KNN“ (5 Seiten)

1 - Quiz		
Was ist die Definition von Intelligenz?		20 Sekunden
	... was der Intelligenztest misst.	✗
	Fähigkeit schwierige Probleme zu lösen	✗
	Fähigkeit abstrakt zu denken	✗
	Es gibt keine allgemeingültige Definition über alle Fachrichtungen hinweg.	✓
2 - Quiz		
Was ist aus wissenschaftlicher Sicht KEINE Art von Intelligenz?		20 Sekunden
	emotional	✗
	tierisch	✓
	motorisch	✗
	sozial	✗
3 - Quiz		
Was kann KI aktuell noch nicht?		20 Sekunden
	Lösen von komplexen Problemen (Reden, Planen, Kreativität, ..)	✓
	Bildererkennung	✗
	komplexe Spiele (z.B. Schach, Go)	✗
	Erlernen von speziellen Aufgaben (z.B. Klassifikation)	✗

4 - Quiz

Wie nennt man generell die Eingabe in der KI?



20 Sekunden

- | | | |
|--|-----------|---|
| | Output | ✗ |
| | Input | ✓ |
| | Data | ✗ |
| | Testdaten | ✗ |

5 - Quiz

Mit welchen Daten wird die Accuracy einer KI geprüft?



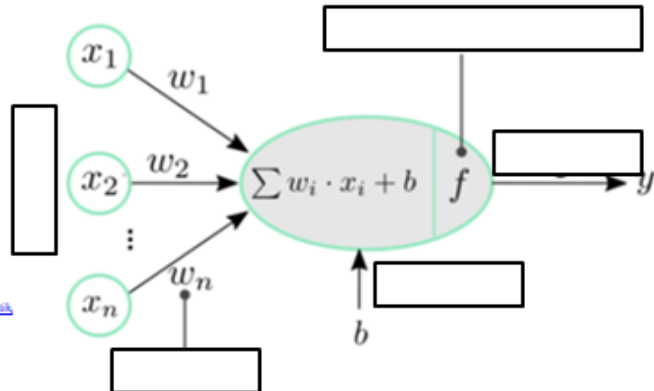
20 Sekunden

- | | | |
|--|----------------|---|
| | Trainingsdaten | ✗ |
| | Testdaten | ✓ |
| | Input | ✗ |
| | Klassifikation | ✗ |

2) Arbeitsblatt „Berechnung KNN“ (5 Seiten)

Aufgabe 1:

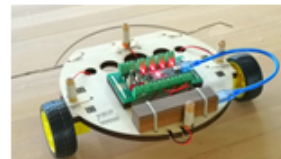
Beschrifte die Zeichnung.



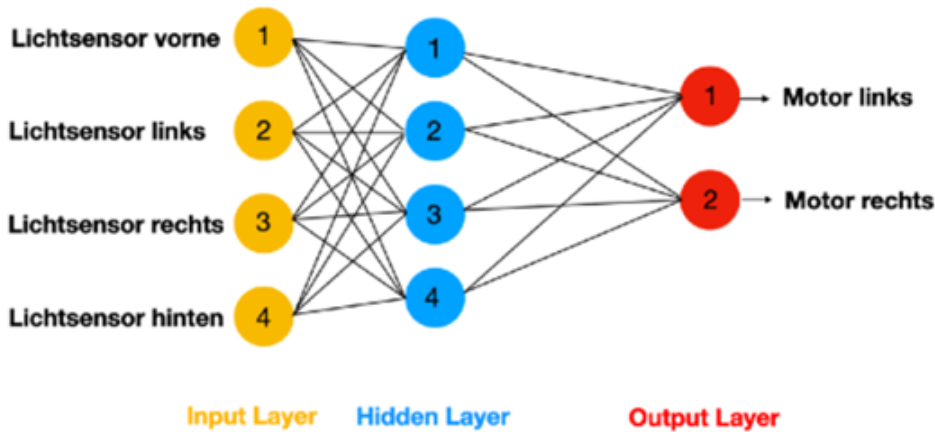
Quelle:
Müller, Jule: Maschinelles Lernen. URL:
<https://fulmsualltagpostloisic3.wordpress.com/informatik/>
(Stand: 31.05.2024).

Aufgabe 2:

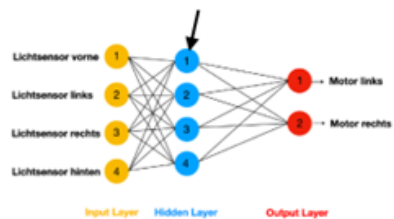
Das künstliche neuronale Netz erhält verschiedene Sensor-Inputs (d.h. Lichtstärken an den Sensoren). Für die Inputs können die zugehörigen Outputs berechnet werden. Im Beispiel hier würden die Outputs für einen zweirädrigen Roboter zur Steuerung der Motorrichtung und Geschwindigkeit verwendet werden. Dabei bedeutet eine positive Zahl, dass das entsprechende Rad den Roboter vorwärts bewegt, während eine negative ihn rückwärts bewegt.



Das KNN des Roboters sieht wie folgt aus (es wurde absichtlich ein KNN mit minimaler Anzahl von Neuronen gewählt):



- Es fällt nur Licht auf den vorderen Sensor. Der Einfachheit halber gibt es nur zwei Stufen: 0 = kein Lichteinfall, 1 = voller Lichteinfall. **Berechnen** Sie, welche Bewegung der Roboter ausführt.
- Erklären** Sie ihr Ergebnis im Sachzusammenhang.



Hidden Neuron 1

Input Neuron 1 (Vo) → $\underline{\hspace{2cm}} \times 1.1 = \underline{\hspace{2cm}}$

+

Input Neuron 2 (Li) → $\underline{\hspace{2cm}} \times 2.6 = \underline{\hspace{2cm}}$

+

Input Neuron 3 (Re) → $\underline{\hspace{2cm}} \times -12.5 = \underline{\hspace{2cm}}$

+

Input Neuron 4 (Hi) → $\underline{\hspace{2cm}} \times -7.6 = \underline{\hspace{2cm}}$

+

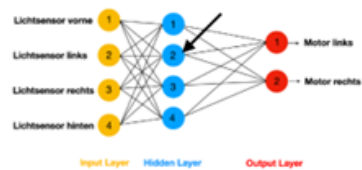
-4.9 (Bias)

=

$\underline{\hspace{2cm}}$

Output nach der ReLU-Aktivierungsfunktion:

$\underline{\hspace{2cm}}$



Hidden Neuron 2

Input Neuron 1 (Vo) → $\underline{\hspace{2cm}} \times -27.7 = \underline{\hspace{2cm}}$

+

Input Neuron 2 (Li) → $\underline{\hspace{2cm}} \times -7.6 = \underline{\hspace{2cm}}$

+

Input Neuron 3 (Re) → $\underline{\hspace{2cm}} \times 7.4 = \underline{\hspace{2cm}}$

+

Input Neuron 4 (Hi) → $\underline{\hspace{2cm}} \times -15.0 = \underline{\hspace{2cm}}$

+

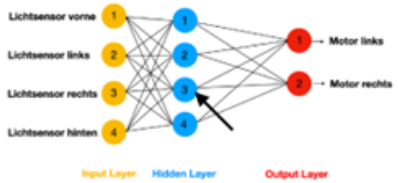
6.5 (Bias)

=

$\underline{\hspace{2cm}}$

Output nach der ReLU-Aktivierungsfunktion:

$\underline{\hspace{2cm}}$



Hidden Neuron 3

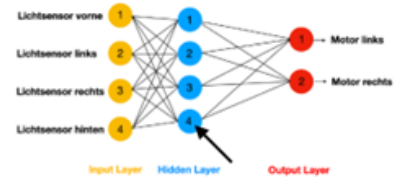
Input Neuron 1 (Vo) → _____ x 27.1 = _____
+

Input Neuron 2 (Li) → _____ x -4.2 = _____
+

Input Neuron 3 (Re) → _____ x -2.9 = _____
+

Input Neuron 4 (Hi) → _____ x -14.1 = _____
+
-9.0 (Bias)
=

Output nach der ReLU-Aktivierungsfunktion:



Hidden Neuron 4

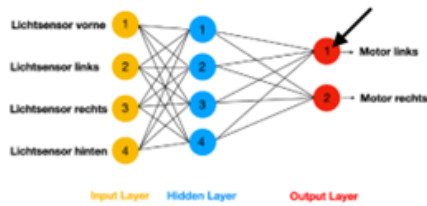
Input Neuron 1 (Vo) → _____ x -14.5 = _____
+

Input Neuron 2 (Li) → _____ x 10.7 = _____
+

Input Neuron 3 (Re) → _____ x -3.2 = _____
+

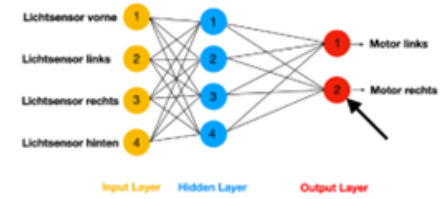
Input Neuron 4 (Hi) → _____ x 9.2 = _____
+
-3.8 (Bias)
=

Output nach der ReLU-Aktivierungsfunktion:



Output Neuron 1

$$\begin{aligned}
 \text{Hidden Neuron 1} &\rightarrow \quad \underline{\hspace{2cm}} \times 19.0 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 2} &\rightarrow \quad \underline{\hspace{2cm}} \times 4.7 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 3} &\rightarrow \quad \underline{\hspace{2cm}} \times 7.4 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 4} &\rightarrow \quad \underline{\hspace{2cm}} \times -10.9 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 &\hspace{10cm} \mathbf{14.4 \text{ (Bias)}} \\
 &\hspace{10cm} = \\
 &\underline{\hspace{2cm}}
 \end{aligned}$$



Output Neuron 2

$$\begin{aligned}
 \text{Hidden Neuron 1} &\rightarrow \quad \underline{\hspace{2cm}} \times 7.1 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 2} &\rightarrow \quad \underline{\hspace{2cm}} \times -5.9 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 3} &\rightarrow \quad \underline{\hspace{2cm}} \times 13.5 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 \text{Hidden Neuron 4} &\rightarrow \quad \underline{\hspace{2cm}} \times -15.6 = \underline{\hspace{2cm}} \\
 &\hspace{10cm} + \\
 &\hspace{10cm} \mathbf{10.6 \text{ (Bias)}} \\
 &\hspace{10cm} = \\
 &\underline{\hspace{2cm}}
 \end{aligned}$$

Zusatzaufgabe:

Es fällt nur Licht auf den hinteren Sensor. Der Einfachheit halber gibt es nur zwei Stufen: 0 = kein Lichteinfall, 1= voller Lichteinfall. **Berechnen** Sie, welche Bewegung der Roboter ausführt.

3) Programmcode „KNN für den Irisdatensatz mit Python“

```
from sklearn import datasets
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
import os
import datetime
import tensorflow as tf
from tensorflow.keras import layers, Sequential
from tensorboard import notebook

#%load_ext tensorboard#d
#%tensorboard --logdir logs

# Das Iris-Datenset wird geladen
iris = datasets.load_iris()
x = iris['data']
y = pd.DataFrame(iris['target']) # We convert the target to a
dataframe
y.columns = ['target'] # We give the dataframe column a name,
here is 'target'
y = pd.get_dummies(y, columns = ['target']) # One hot encode the
column
y = np.array(y) # convert it back to array
x_train_val, x_test, y_train_val, y_test = train_test_split(x,
y, test_size=0.1, random_state=42)
x_train, x_val, y_train, y_val = train_test_split(x_train_val,
y_train_val, test_size=0.1, random_state=42)

# Standardisierung (Werte sollen zwischen -1 und 1 liegen)
train_avg = x_train.mean(axis=0)
train_std = x_train.std(axis=0)
x_train = (x_train - train_avg)/train_std
val_avg = x_val.mean(axis=0)
```



```

val_std = x_val.std(axis=0)
x_val = (x_val - val_avg)/val_std
test_avg = x_test.mean(axis=0)
test_std = x_test.std(axis=0)
x_test = (x_test - test_avg)/test_std

# Model des KNN wird aufgebaut
model = Sequential([
    layers.Dense(10, input_dim=4, activation='relu'),
    layers.Dense(10, activation='relu'), # can be changed
    layers.Dense(5, activation='relu'), # can be changed
    layers.Dense(4, activation='relu'), # can be changed
    layers.Dense(3, activation='softmax') # don't change this
one
])

# Model des KNN wird kompiliert
model.compile(
    optimizer='rmsprop', # find out about different optimizers
and try to use them Try using Adam optimizer with lr(learning
rate)
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

#logdir = os.path.join("logs",
datetime.datetime.now().strftime("%Y%m%d-%H%M%S"))
#tensorboard_callback = tf.keras.callbacks.TensorBoard(logdir,
histogram_freq=1)

# das KNN wird trainiert
model.fit(
    x_train,
    y_train,

```

```
    epochs=100, # can change the epochs
    batch_size=5, # can change the batch size
    validation_data=(x_val, y_val),
    # callbacks=[tensorboard_callback] # This is when tensorflow
send the results back to the tensorboard
)

# notebook.list() # View open TensorBoard instances

#notebook.display(port=6006, height=1000)

# Das KNN wird getestet
model.evaluate(x_test, y_test)
```

4) Arbeitsblätter „Grenzen von Optimierung mit KI an Fisher`s Iris-Dataset“(1. Aufbau des KNN, 2. Lernrate, 3. Overfitting)

Aufgabe zum Aufbau des KNN

Daten zum Training Ihres KNN:

Training-Validation-Test: 121-14-15 Epochen: 50 Learning-Rate: 0,001 Batch-Size: 5

#	Ergebnisse	KNN Aufbau	Accuracy	Loss
1	<pre>model = Sequential([layers.Dense(0, input_dim=4, activation='relu'), layers.Dense(3, activation='softmax')]) model.evaluate(x_test, y_test) 1/1 [*****] - @s 33ms/step - loss: 1.0986 - accuracy: 0.4000</pre>			
2	<pre>model = Sequential([layers.Dense(5, input_dim=4, activation='relu'), layers.Dense(3, activation='softmax')]) model.evaluate(x_test, y_test) 1/1 [*****] - @s 25ms/step - loss: 0.3011 - accuracy: 0.8000</pre>			
3	<pre>model = Sequential([layers.Dense(10, input_dim=4, activation='relu'), layers.Dense(3, activation='softmax')]) model.evaluate(x_test, y_test) 1/1 [*****] - @s 145ms/step - loss: 0.2451 - accuracy: 0.8667</pre>			
4	<pre>model = Sequential([layers.Dense(10, input_dim=4, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(3, activation='softmax')]) model.evaluate(x_test, y_test) 1/1 [*****] - @s 225ms/step - loss: 0.3077 - accuracy: 0.8667</pre>			
5	<pre>model = Sequential([layers.Dense(10, input_dim=4, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(10, activation='relu'), layers.Dense(3, activation='softmax')]) model.evaluate(x_test, y_test) 1/1 [*****] - @s 31ms/step - loss: 0.6466 - accuracy: 0.8000</pre>			

Erklären Sie den Zusammenhang zwischen Aufbau des neuronalen Netzes und der Accuracy. Notieren Sie dazu den Aufbau des Netzes in den rechten Spalten (z.B. 3-5-4).



Zusatzaufgabe: nächste Seite



Zusatzaufgaben:

1. Finden Sie heraus, was das Gesetz „Ockhams razor“ bedeutet und erklären Sie, wie das zu Ihren Ergebnissen passt.

2. Bereiten Sie eine Kurzpräsentation (ca. 1 Min) Ihrer Ergebnisse vor. Als Unterstützung wird diese Power Point Folie eingeblendet sein:

AUFBAU DES KNN

Ockham's Razor:

¹ von links Janssen (o.D.) Singh (2023)

Aufgabe zur Lernrate

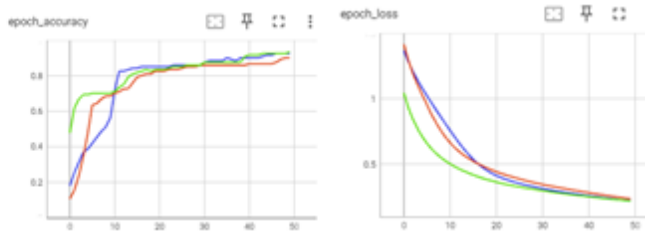
Daten zum Training Ihres KNN:

Training-Validation-Test: 121-14-15

Epochen: 50

KNN: 4-10-3

Batch-Size: 5



1: blau 2: orange 3: grün

#	Ergebnisse	Learning Rate	Accuracy	Loss
1	<pre>adam = Adam(learning_rate=0.001) model.compile(optimizer='adam', model.evaluate(x_test, y_test) 1/1 [=====] - 0s 24ms/step - loss: 0.2830 - accuracy: 0.8667</pre>			
2	<pre>adam = Adam(learning_rate=0.1) model.compile(optimizer='adam', model.evaluate(x_test, y_test) 1/1 [=====] - 0s 25ms/step - loss: 0.2682 - accuracy: 0.8000</pre>			
3	<pre>adam = Adam(learning_rate=1.0) model.compile(optimizer='adam', model.evaluate(x_test, y_test) 1/1 [=====] - 0s 23ms/step - loss: 0.3019 - accuracy: 0.8000</pre>			

Füllen Sie die rechten Spalten aus. Erklären Sie anhand der Ergebnisse den Zusammenhang zwischen Lernrate und der Accuracy:



Zusatzaufgabe: nächste Seite



Zusatzaufgabe:

1. Ziel: Minimaler Fehler $F(\text{Loss})$ mit Backpropagation \rightarrow das globale Minimum erreichen



Abbildung 1 "Zu kleine Lernrate"

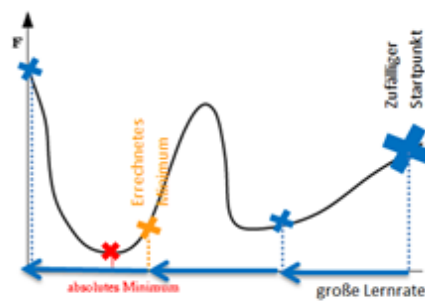


Abbildung 2 "Zu große Lernrate"

Erklären Sie die Problematik zu kleiner und zu großer Lernrate an den Abbildungen 1 und 2.

2. Bereiten Sie eine Kurzpräsentation (ca. 1 Min) Ihrer Ergebnisse vor. Als Unterstützung wird diese Power Point Folie eingeblendet sein:

LERNRATE

Die Folie zeigt zwei Diagramme, die die Auswirkungen von kleiner und großer Lernrate auf den Fehler F über die Iterationen illustrieren. Das linke Diagramm zeigt eine kleine Lernrate, bei der der Fehler langsam zum globalen Minimum konvergiert. Das rechte Diagramm zeigt eine große Lernrate, bei der der Fehler schnell zum globalen Minimum konvergiert, aber auch das Überschreiten des globalen Minimums und das Erreichen eines neuen lokalen Minimums möglich ist.

Aufgabe zum Overfitting

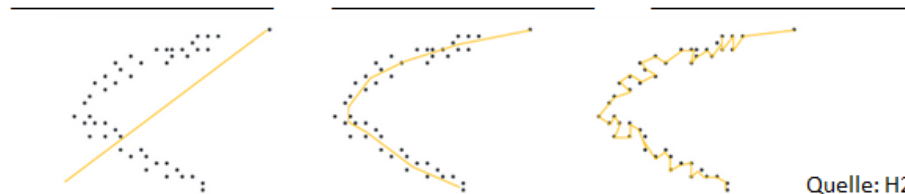
Daten zum Training Ihres KNN:

Training-Validation-Test: 121 -14- 15 Lernrate: 0,001 Epochen: 120 KNN: 4-10-3 Batch-Size: 1

Epochen	20	40	60	120
Ergebnisse blau = Training (T) rot = Validation (V)	<p>epoch_loss tag: epoch_loss</p>	<p>epoch_loss tag: epoch_loss</p>	<p>epoch_loss tag: epoch_loss</p>	<p>epoch_loss tag: epoch_loss</p>
	Loss T= Loss V=	Loss T= Loss V=	Loss T= Loss V=	Loss T= Loss V=

1. Füllen Sie Zeile 3 aus. Beschreiben Sie den Verlauf der Kurven des Verlustes bei den Trainings- und den Validationsdaten über die Epochen hinweg.

2. Ordnen Sie den Bildern folgende Begriffe zu:
Overfitting, Underfitting und Good fit zu:



Quelle: H2O.ai (o.D.)

Overfitting:

Overfitting bedeutet übersetzt Überanpassung und bezeichnet die Überanpassung von Modellen bzw. deren Parameter an einen Datensatz. Das Modell lernt den Datensatz „auswendig“, erkennt aber nicht das zugrundeliegende Muster oder System. Damit können unbekannte Daten nicht gut vorhergesagt werden. In der Praxis wird dieses Phänomen aber für die Technik Early Stopping verwendet. Die optimierende Person lässt das Training einmal mit Overfitting durchlaufen. Dann sieht sie, in welcher Epoche das Modell overfitted. Den nächsten Durchlauf lässt sie nur bis zu dieser Epoche laufen, weil sie hier die besten Ergebnisse hat. Mit dieser Epochenzahl optimiert sie die anderen Parameter ihres Modells.

Wenn Overfitting unerwünscht ist, kann man es mit einem möglichst großen Datensatz („Auswendiglernen“ wird schwieriger) oder durch immer durchtauschende Trainings- und Validationsdaten verhindern. Den Bereich, bevor ein Modell optimiert ist nennt man übrigens Underfittings, das heißt im „normalen“ Training underfitted ein Modell die ganze Zeit, so lange, bis es optimiert ist oder overfitted.

3. In welchen Epochen finden in unserem Beispiel folgende Phänomene statt:

Phänomen	Underfitting	Good Fit	Overfitting
Epoche			

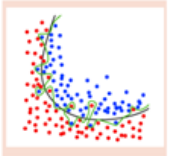
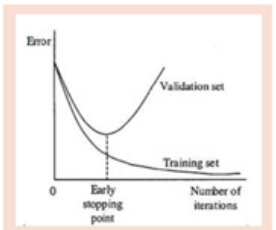

4. Erläutern Sie das Phänomen Overfitting in eigenen Worten anhand der Diagramme von Aufgabe 1.

**Zusatzaufgaben:**

1. Nennen Sie zwei Methoden, mit denen das Risiko von Overfitting vermieden werden kann? _____

2. Bereiten Sie eine Kurzpräsentation (ca. 1 Min) Ihrer Ergebnisse vor.
Als Unterstützung wird diese Power Point Folie eingeblendet sein:

OVERFITTING

2 von

(Gb)